Optimal Exploitation of Clustering and History Information in Multi-armed Bandit Problem

Djallel Bouneffouf [1] , **Srinivasan Parthasarathy** [1] , **Horst Samulowitz** [1] , **Martin Wistuba** [1]

Abstract

We consider the stochastic multi-armed bandit problem and the contextual bandit problem with historical observations and pre-clustered arms. The historical observations can contain any number of instances for each arm, and the pre-clustering information is a fixed clustering of arms provided as part of the input. We develop a variety of algorithms which incorporate this offline information effectively during the online exploration phase and derive their regret bounds. In particular, we develop the META algorithm which effectively hedges between two other algorithms: one which uses both historical observations and clustering, and another which uses only the historical observations. The former outperforms the latter when the clustering quality is good, and vice-versa. Extensive experiments on synthetic and real world datasets on Warafin drug dosage and web server selection for latency minimization validate our theoretical insights and demonstrate that META is a robust strategy for optimally exploiting the pre-clustering information.

1. Introduction

Many sequential decision problems ranging from clinical trials to online ad placement can be modeled as multi-armed bandit problems (classical bandit) (Li et al., 2010a; Tang et al., 2013). At each time step, the algorithm chooses one of several possible actions and observes its reward with the goal of maximizing the cumulative reward over time. A useful extension to classical bandit is the contextual multi-arm bandit problem, where before choosing an arm, the algorithm observes a context vector in each iteration (Langford and Zhang, 2007; Agrawal and Goyal, 2013). In the conventional formulation of these problems, arms are assumed to be unrelated to each other and no prior knowledge about the arms exist. However, applications can often provide historical observations about arms and also similarity information between them prior to the start of the online exploration phase. In this work, we assume that similarity between the arms is given in the form of a fixed pre-clustering of arms and we design algorithms which exploit the historical information and the clustering information opportunistically. In particular, we seek algorithms which satisfy the following property: if the quality of clustering is good, the algorithm should quickly learn to use this information aggressively during online exploration; however, if the quality of clustering is poor, the algorithm should ignore this information. We note at the outset that it is possible to consider alternative formulations of the bandit problem in the presence of history and clustering information. For instance, one alternative is to use clustering metrics such as Dunn or Dunn-like indices (Liu et al., 2010) to decide if the clustering information should be used during online exploration. However, in general, cluster validation metrics are tightly coupled with specific classes of clustering algorithms (e.g., distance vs. density based clustering); hence, we focus on bandit algorithms which are agnostic to specific clustering metrics and work with *any given* clustering of the arms. Other alternatives include the design of algorithms for optimal pre-clustering of arms prior to online exploration, as well as incrementally modifying the clustering of the arms during online exploration. These are both valuable refinements to the problem we study but are beyond the scope of the current work. In this paper, we focus on the problem of *optimal exploitation of the clustering information given as part of the input*.

A real-world motivation for our work is the problem of dosing the drug Warfarin (Sharabiani et al., 2015). Correctly dosing Warfarin is a significant challenge since it is dependent on the patient's clinical, demographic and genetic information. In the contextual bandit setting, this information can be modeled as the context vector and the bandit arms model the various dosage levels. The dosage levels are further grouped into distinct clusters (15 arms and 3 clusters in our dataset). Historical treatment responses for various dosage levels as well as clustering information derived from medical domain knowledge is available as part this problem instance. Another motivating application is web server selection for latency minimization in content distribution networks (CDNs)(Krishnamurthy, Wills, and Zhang, 2001a). A CDN can choose mirrored content from several distributed web servers. The latency of these servers are correlated and vary widely due to geog-

*Equal contribution [1]IBM Research, Yorktown Heights, NY, USA. Correspondence to: Djallel Bouneffouf <djallel.bouneffouf@ibm.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

raphy and network traffic conditions. This domain can be modeled as the classical bandit problem with the bandit arms representing various web servers, and latency being the negative reward. The web servers are further grouped into clusters based on their historical latencies (700 arms and 5 clusters in our dataset). These problem domains share the characteristics of the availability of historical observations to seed the online phase and grouping of arms into clusters either through domain knowledge or through the use of clustering algorithms.

2. Related Work

Both the classical multi-armed bandit and the contextual bandit problems have been studied extensively along with their variants (Lai and Robbins, 1985; Auer, Cesa-Bianchi, and Fischer, 2002; Thompson, 1933; Kaufmann, Korda, and Munos, 2012; Auer and Cesa-Bianchi, 1998; Auer et al., 2002; Maillard and Mannor, 2014; Gentile, Li, and Zappella, 2014; Nguyen and Lauw, 2014; Korda, Szörényi, and Li, 2016; Gentile et al., 2017; Pandey, Chakrabarti, and Agarwal, 2007; Shivaswamy and Joachims, 2012). The works which are closely related to ours are (Pandey, Chakrabarti, and Agarwal, 2007) and (Shivaswamy and Joachims, 2012). (Pandey, Chakrabarti, and Agarwal, 2007; Wang, Zhou, and Shen, 2018) study the classical bandit problem under the same model of arm clustering as in this work, and (Shivaswamy and Joachims, 2012) studies the classical bandit problem under the same model of historical observations as in this work. In contrast to (Pandey, Chakrabarti, and Agarwal, 2007; Wang, Zhou, and Shen, 2018; Shivaswamy and Joachims, 2012), our work provides 1) algorithms which *simultaneously* incorporate historical observations and clustering information in the classical bandit setting, 2) regret guarantees under tight and adversarial clustering for this setting, 3) algorithms which *simultaneously* incorporate historical observations and clustering information in the contextual bandit setting; we also provide regret guarantees for our classical bandit algorithm which uses history; prior to this work, we are not aware of such extensions for the contextual bandit setting, and 4) the META algorithm which effectively hedges between the strategy that uses both clustering and historical observations vs. the strategy which uses only the historical observations and not the clustering information.

3. Problem Setting

Classical Bandit: The classical bandit problem is defined as exploring the response of K arms within T trials. Playing an arm yields an immediate, independent stochastic reward according to some fixed unknown distribution associated with the arm whose support is in $(0,1)$. The task is to find the reward-maximizing policy.

We adapt this scenario by assuming that the arms are grouped into C clusters with arm k assigned to cluster $c(k)$. Unlike the classical setting, we also wish to incorporate historical observations which may be available for the arms. Specifically, for $t \in \{1,2,...,T\}$, let $r_k(t)$ denote the online reward from arm k at time t. For notational convenience, we assume it to be 0 if k was not played at time t. Let $r_k^h(t)$ denote the historical reward for the t^{th} instance of playing arm k in history. H_k is the number of historical instances available for arm k, and H is defined as $H := \sum_{k=1}^{K} H_k$. For each arm, the historical rewards are drawn independently from the same distribution as the online rewards.

Let θ_k denote the expected reward for arm k. The goal is to maximize the expected total reward during T iterations $\mathbb{E}\left[\sum_{t=1}^{T} \theta_{k(t)}\right]$, where $k(t)$ is the arm played in step t, and the expectation is over the random choices of $k(t)$ made by the algorithm. An equivalent performance measure is the expected total regret, which is the amount of total reward lost by a specific algorithm compared to an oracle which plays the (unknown) optimal arm during each step. The expected total regret is defined as:

$$\mathbb{E}[R(T)] \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=1}^{T} \left(\theta^* - \theta_{k(t)}\right)\right] = \sum_{k} \Delta_k \mathbb{E}[n_k(T)] \qquad (1)$$

with $\theta^* \stackrel{\text{def}}{=} \max_k \theta_k$, $\Delta_k \stackrel{\text{def}}{=} \theta^* - \theta_k$, and $n_k(t)$ denotes the number of times arm k has been played until t.

Contextual Bandit: The contextual bandit with history and pre-clustered arms is defined as follows: At each step $t \in \{1,...,T\}$, the player is presented with a *context* (*feature vector*) $\vec{x}_t \in \mathbb{R}^d$ before playing an arm from the set $A = \{1,...,N\}$ that are grouped into C known clusters. Let $r_k(t)$ denote the reward which the player can obtain by playing arm k at time t given context $\vec{x}_t$. As in (Chu et al., 2011), we will primarily focus on bandits where the expected reward of an arm is linearly dependent on the context. Specifically, $\forall k, t : r_k(t) \in [0,1]$ and $\mathbb{E}[r_k(t)|\vec{x}_t] = \theta_k^\top \vec{x}_t$ where $\theta_k \in \mathbb{R}^d$ is an unknown coefficient vector associated with the arm k which needs to be learned from data.

To incorporate historical information about rewards of the arms in a principled manner, we assume that each arm k is associated with a matrix $\mathbf{H}_k \in \mathbb{R}^{d \times d}$ which records covariance information about the contexts played in history during times t with $1 \le t \le H_k$. $\mathbf{H}_k$ is computed using all observed context $\vec{x}_t$ so that:

$$\mathbf{H}_k(t) = \mathbf{H}_k(t-1) + \vec{x}_t \vec{x}_t^\top, \quad \mathbf{H}_k(0) = \mathbf{I}_d, \mathbf{H}_k = \mathbf{H}_k(H_k)$$

Here, $\mathbf{I}_d$ is the identity matrix of dimension d. The arm is also associated with a vector parameter b_k^h which is the weighted sum of historical contexts, where the weights are the respective rewards:

$$b_k^h(t) = b_k^h(t-1) + r_k^h(t)\vec{x}_t, \qquad b_k^h(0) = \mathbf{0}$$

The role of the covariance matrix in HLINUCBC is analogous to its role in LINUCB which in turn is analogous to its role in least squares linear regression.

4. HUCBC for Classical Bandit

One solution for the classical bandit problem is the well known Upper Confidence Bound (UCB) algorithm (Auer,

Cesa-Bianchi, and Fischer, 2002). This algorithm plays the current best arm at every time step, where the best arm is defined as one which maximizes the sum of observed mean rewards and an uncertainty term. We adapt UCB for our new setting such that it can incorporate both clustering and historical information. Our algorithm incorporates the historical observations by utilizing it both in the computation of the observed mean rewards and the uncertainty term. Our algorithm incorporates the clustering information by playing at two levels: first picking a cluster using a UCB-like strategy at each time step, and subsequently picking an arm within the cluster, again using a UCB-like strategy.

The pseudocode for HUCBC is presented in Algorithm 1 and the mean reward for arm k is defined in (2).

$$\hat{\theta}_k^h(t) = \frac{\sum_{t'=1}^{H_k} r_k^h(t') + \sum_{t'=1}^{t} r_k(t')}{H_k + n_k(t)} \tag{2}$$

$$\mathrm{HUCBC}_k(t) = \hat{\theta}_k^h(t) + \sqrt{\frac{2log(t+H_k)}{n_k(t)+H_k}} \tag{3}$$

At each time step t of the exploration, HUCBC computes the quantity $\mathrm{HUCBC}_k(t)$ for arm k using (3). $\mathrm{HUCBC}_k(t)$ represents an *optimistic* estimate of the reward attainable by playing arm k: this estimate incorporates the mean reward observed for the arm so far including the plays of the arm in history and an upper confidence term to account for the possibility of underestimation due to randomness in the rewards. In a similar manner, HUCBC also computes $\hat{\theta}_i^{hc}(t)$ and $\mathrm{HUCBC}_i^c(t)$ for each cluster i:

$$\hat{\theta}_i^{hc}(t) = \frac{\sum_{t'=1}^{H_i^c} r_i^{hc}(t') + \sum_{t'=1}^{t} r_i^c(t')}{H_i^c + n_i^c(t)} \tag{4}$$

$$\mathrm{HUCBC}_i^c(t) = \hat{\theta}_i^{hc}(t) + \sqrt{\frac{2log(t+H_i^c)}{n_i^c(t)+H_i^c}} \tag{5}$$

Note that $\hat{\theta}_i^{hc}(t)$ is the mean reward observed for cluster i including plays of the cluster in history and $\mathrm{HUCBC}_i^c(t)$ represents an *optimistic* estimate of the reward attainable by playing cluster i. The quantities $r_i^{hc}(t)$, $r_i^c(t)$, $n_i^c(t)$, and H_i^c in (4) and (5) are the per-cluster analogues of the corresponding quantities defined per-arm. Also note that, while the per-cluster quantities carry a superscript c as part of their notation, the per-arm quantities do not. At each time step t, HUCBC first picks the cluster which maximizes $\mathrm{HUCBC}_i^c(t)$ and then picks the arm within this cluster which maximizes $\mathrm{HUCBC}_k(t)$.

Algorithm 1 The HUCBC algorithm

At time t, select cluster i that maximizes $\mathrm{HUCBC}_i^c(t)$ in (5) and play arm k in cluster i that maximizes $\mathrm{HUCBC}_k(t)$ in (3)

Regret Analysis: We upper bound the expected number of plays of a sub optimal cluster under tight clustering of arms. Let i^* denote the cluster containing the best arm. Arms are said to be tightly clustered if: 1) for each cluster i, there exists an interval $[l_i, u_i]$ which contains the expected reward of every arm in cluster i, and 2) for $i \neq i^*$, the intervals $[l_i, u_i]$ and $[l_{i^*}, u_{i^*}]$ are disjoint. For any cluster $i \neq i^*$, define $\delta_i = l_{i^*} - u_i$.

Theorem 1 (Tight Clustering)**.** *The expected regret $\mathbb{E}[R(T)]$ at any time horizon T under tight clustering of the arms in HUCBC is at most the following:*

$$\sum_{i \neq i^*} \left(1 + \ell_i^c + \frac{\pi^2(1+6H_i^c)}{6(2H_i^c+1)^2} + \frac{\pi^2(1+6H_{i^*}^c)}{6(2H_{i^*}^c+1)^2}\right) + \sum_{k|c(k)=i^*} \left(1 + \ell_k + \frac{\pi^2(1+6H_k)}{6(2H_k+1)^2} + \frac{\pi^2(1+6H_{k^*})}{6(2H_{k^*}+1)^2}\right) \tag{6}$$

Here, $\ell_i^c = \max\left(0, \frac{8\log(T+H_i^c)}{\delta_i^2} - H_i^c\right)$, $\ell_k = \max\left(0, \frac{8\log(T+H_k)}{\Delta_k^2} - H_k\right)$ *and* $k^* = \mathrm{argmax}_k \theta_k$.

Intuition behind Theorem 1: First, the cumulative regret up to time T is logarithmic in T. Second, as the separation between clusters increases resulting in increased Δ_k values, the regret decreases. Third and a somewhat subtle aspect of this regret bound is that the number of terms in the summation equals the number of arms in the optimal cluster + the number of sub-optimal clusters. This number is always upper bounded by the total number of arms. In fact, the difference between them can be pronounced when the clustering is well balanced – for example, when there are $\sqrt{n}$ clusters with $\sqrt{n}$ arms each, the total number of terms in the summation is $2\sqrt{n}$ while the total number of arms is n. When the quality of clustering is good, this is exactly the aspect of the regret bound which tilts the scales in favor of HUCBC compared to UCB or HUCB.

Proof. The proof consists of three steps whose ideas are as follows. In the first step, we decompose the HUCBC regret into two parts: regret contributed by sub-optimal clusters (i.e., clusters without the optimal arm) and regret contributed by the sub-optimal arms within the optimal cluster (i.e, the cluster containing the optimal arm). The latter quantity can be upper bounded using known results for HUCB. The main challenge in our analysis is to upper bound the former, which we accomplish in Step 2. The main idea in Step 2 is to develop a concentration inequality for the reward obtained from any cluster. We accomplish this by proving that the cumulative reward deviation (i.e., the cumulative sum of actual rewards minus their expectation) is a martingale. In the analysis of UCB and HUCB, a concentration inequality of this type is derived through the use of Chernoff bounds: these are not applicable in our analysis due to the fact that rewards from a cluster are not only non-stationary but also not independent and highly correlated with rewards from previous plays of the cluster. We circumvent this difficulty through the use of martingale concentration inequalities in place of Chernoff bounds. In Step 3, we tie the results of Steps 1 and 2 together to obtain our final regret bound for HUCBC. The formal proof is as follows. **Step 1:** Since HUCBC uses HUCB to play arms within each cluster, we start by introducing the following regret bound for HUCB (Shivaswamy and Joachims, 2012).

Fact 2 (Theorem 2, (Shivaswamy and Joachims, 2012))**.** *The expected number of plays of any sub optimal arm k, within any cluster i, for any time horizon T, for any clustering of arms, satisfies:*

$$\mathbb{E}[n_k(T)] \leq 1 + \ell_k + \frac{\pi^2(1+6H_k)}{6(2H_k+1)^2} + \frac{\pi^2(1+6H_{k^*})}{6(2H_{k^*}+1)^2} \tag{7}$$

Here k^* is the best arm within cluster i, and $\ell_k = \max\Big(0, \frac{8\log(T+H_k)}{\nabla_k^2} - H_k\Big)$, where $\nabla_k = \theta_{k^*} - \theta_k$.

For any time horizon, the number of plays of a sub optimal arm in HUCBC is an upper bound on its regret. This quantity equals the sum of the number of plays of a sub optimal cluster along with the number of plays of sub optimal arms within the optimal cluster. The latter quantity is upper bounded using Fact 2 since for any sub optimal arm k in the optimal cluster i^*, $\Delta_k = \nabla_k$. The former quantity is upper bounded in **Steps 2** and **3**. **Step 2:** Given cluster i, let t_ℓ be the time slot when cluster i is played for the ℓ^{th} time in the online phase. Suppose $t_1, t_2, ...$ be fixed and given. Define:

$$z_i^c(\ell) = \begin{cases} 0 & \text{if } \ell = 0 \\ z_i^c(\ell-1) + r_i^{hc}(\ell) - \theta_{j^h(\ell)} & \text{if } 1 \le \ell \le H_i^c \\ z_i^c(\ell-1) + r_i^c(\ell - H_i^c) - \theta_{j(\ell - H_i^c)} & \text{if } \ell > H_i^c \end{cases}$$

In this definition, $j^h(\ell)$ is the arm that was played during the ℓ^{th} instance of cluster i in the historical phase, and $j(\ell)$ is the arm that is played during the ℓ^{th} instance of cluster i in the online phase, r (with the appropriate subscripts and superscripts) refers to the actual reward obtained for that instance, and θ (with the appropriate subscripts and superscripts) refers to the mean reward of the arm for that instance. We claim that the random sequence $\{z_i^c(\ell) | \ell = 1, 2, ...\}$ is a martingale. Indeed, for $1 \le \ell \le H_i^c$, we have:

$$\mathbb{E}[z_i^c(\ell) | z_i^c(\ell-1) = z] = z + \mathbb{E}[r_i^{hc}(\ell) - \theta_{j^h(\ell)} | z_i^c(\ell-1) = z]$$
$$= z + \sum_{j | c(j) = i} \Pr[j^h(\ell) = j | z_i^c(\ell-1) = z]$$
$$\mathbb{E}[r_j^h(\ell) - \theta_j | z_i^c(\ell-1) = z \bigwedge j^h(\ell) = j] = z$$

A similar argument holds when $\ell > H_i$ which implies $\{z_i^c(\ell) | \ell = 1, 2, ...\}$ is a martingale. It is easy to verify that $\forall \ell,\ |z_i^c(\ell) - z_i^c(\ell-1)| < 1$; hence, by Azuma-Hoeffding inequality (Alon and Spencer, 2016), we have:

$$\Pr[|z_i^c(\ell)| \ge v] = \Pr[|z_i^c(\ell) - z_i^c(0)| \ge v] \le 2e^{-\frac{v^2}{2\ell}} \tag{8}$$

Consider $i = i^*$. We note that since $t_1, t_2, ...$ are fixed, $\forall t$, $n_i^c(t)$ is a fixed number. We have:

$$(8) \Longrightarrow \forall t\colon \Pr[HUCBC_{i^*}^c(t) \le l_{i^*}]$$
$$= \Pr[(n_{i^*}^c(t) + H_{i^*}^c) HUCBC_{i^*}^c(t) \le (n_{i^*}^c(t) + H_{i^*}^c) l_{i^*}]$$
$$\le \Pr\Bigg[\sum_{\ell=1}^{H_{i^*}^c} r_{i^*}^{hc}(\ell) + \sum_{\ell=1}^{n_{i^*}^c(t)} r_{i^*}^c(\ell) + \sqrt{2(n_{i^*}^c(t) + H_{i^*}^c)\log(t + H_{i^*}^c)}$$
$$\le \sum_{k | c(k) = i^*} (n_k(t) + H_k)\theta_j \Bigg]$$
$$= \Pr\Big[z_{i^*}^c(n_{i^*}^c(t) + H_{i^*}^c) \le -\sqrt{2(n_{i^*}^c(t) + H_{i^*}^c)\log(t + H_{i^*}^c)}\Big]$$
$$\overset{(8)}{\le} \frac{2}{(t + H_{i^*}^c)^2} \tag{9}$$

We note that since (9) holds conditionally for any fixed sequence $t_1, t_2, ...$, it holds unconditionally as well. Now consider $i \ne i^*$, and any fixed t such that $n_i^c(t) = \gamma$, where $\gamma \ge \ell_i^c \overset{\text{def}}{=} \max\Big\{0, \frac{8\log(t+H_i^c)}{\delta_i^2} - H_i^c\Big\}$.

$$(8) \Longrightarrow \Pr\Big[HUCBC_i^c(t) \ge u_i + \delta_i \Big| n_i^c(t) = \gamma\Big]$$
$$= \Pr\Big[(\gamma + H_i^c) HUCBC_i^c(t) \ge (\gamma + H_i^c)(u_i + \delta_i) \Big| n_i^c(t) = \gamma\Big] \le$$
$$\Pr\Bigg[\sum_{\ell=1}^{H_i^c} r_i^{hc}(\ell) + \sum_{\ell=1}^{\gamma} r_i^c(\ell) + \sqrt{2(\gamma + H_i^c)\log(t + H_i^c)}$$
$$\ge \sum_{k | c(k) = i} (H_k + n_k(t))\theta_k + (\gamma + H_i^c)\delta_i \Big| n_i^c(t) = \gamma\Bigg]$$
$$= \Pr\Bigg[z_i^c(\gamma + H_i^c) \ge (\gamma + H_i^c)\delta_i - \sqrt{2(\gamma + H_i^c)\log(t + H_i^c)}$$
$$\Big| n_i^c(t) = \gamma\Bigg] \overset{(8)}{\le} \frac{2}{(t + H_i^c)^2} \tag{10}$$

We note that since (10) holds conditionally for any fixed sequence $t_1, t_2, ...$, we have:

$$\Pr\Big[HUCBC_i^c(t) \ge u_i + \delta_i \Big| n_i^c(t) \ge \ell_i^c\Big] \le \frac{2}{(t + H_i^c)^2} \tag{11}$$

Step 3: Suppose we have an event $\mathcal{A}$ such that $\mathcal{A} \Longrightarrow \mathcal{B} \vee \mathcal{C} \vee \mathcal{D}$. Then, for any event $\mathcal{E}$, we have:

$$\mathcal{A} \Longrightarrow (\mathcal{A} \wedge \mathcal{E}) \vee (\mathcal{A} \wedge \overline{\mathcal{E}}) \Longrightarrow (\mathcal{A} \wedge \mathcal{E}) \vee ((\mathcal{B} \vee \mathcal{C} \vee \mathcal{D}) \wedge \overline{\mathcal{E}})$$
$$\text{Hence, } \Pr[\mathcal{A}] \le \Pr[\mathcal{B} \wedge \overline{\mathcal{E}}] + \Pr[\mathcal{C} \wedge \overline{\mathcal{E}}] + \Pr[\mathcal{A} \wedge \mathcal{E}] + \Pr[\mathcal{D} \wedge \overline{\mathcal{E}}]$$
$$\le \Pr[\mathcal{B}] + \Pr[\mathcal{C}] + \Pr[\mathcal{A} | \mathcal{E}] + \Pr[\mathcal{D} | \overline{\mathcal{E}}]$$

Consider a suboptimal cluster i. If $H_i^c = 0$, then HUCB (like UCB) will initialize itself by playing cluster i once at the start of the online phase. Define this as event $\mathcal{B}$. Define $\mathcal{A} = \mathbf{1}(i(t) = i)$, $\mathcal{C} = \mathbf{1}(HUCBC_{i^*}^c(t) \le l_{i^*})$, $\mathcal{D} = \mathbf{1}(HUCBC_i^c(t) \ge u_i + \delta_i)$, and $\mathcal{E} = \mathbf{1}\left(n_i(t) < \ell_i^c\right)$ above. We now have:

$$\mathbb{E}[n_i^c(T)] = \sum_{t=1}^{T} \Pr[i(t) = i] \le 1 + \sum_{t=1}^{T} \Pr[HUCBC_{i^*}^c(t) \le l_{i^*}]$$
$$+ \sum_{t=1}^{T} \Pr\Big[i(t) = i \Big| n_i^c(t) < \ell_i^c\Big] + \sum_{t=1}^{T} \Pr\Big[HUCBC_i^c(t) \ge u_i + \delta_i \Big| n_i^c(t) \ge \ell_i^c\Big]$$
$$\overset{(9) \text{ and } (11)}{\le} 1 + \ell_i^c + \sum_{t=1}^{T} \frac{2}{(t + H_i^c)^2} + \sum_{t=1}^{T} \frac{2}{(t + H_{i^*}^c)^2}$$
$$\le 1 + \ell_i^c + \frac{\pi^2(1 + 6H_i^c)}{6(2H_i^c + 1)^2} + \frac{\pi^2(1 + 6H_{i^*}^c)}{6(2H_{i^*}^c + 1)^2} \tag{12}$$

The theorem now follows by combining (12) and (7). □

We now upper bound the expected number of plays of a sub optimal cluster where an adversary is free to cluster the arms in order to elicit the worst case behavior from HUCBC. For ease of analysis, we will analyze a variant of HUCBC which we denote as $HUCBC'$, which plays UCB at the inter-cluster level and plays HUCB at the intra-cluster level.

Theorem 3 (Adversarial Clustering)**.** *The expected regret at any time horizon T under any clustering of the arms in $HUCBC'$*

satisfies the following:

$$\mathbb{E}[R(T)] \leq \sum_{i \neq i^*} \max_{k|c(k)=i} \left(\frac{16 r \log T}{(\Delta_k/2)^2} + 2s + \frac{\pi}{3} \right) + \sum_{k|c(k)=i^*} \left(1 + \ell_k + \frac{\pi^2(1+6H_k)}{6(2H_k+1)^2} + \frac{\pi^2(1+6H_{k^*})}{6(2H_{k^*}+1)^2} \right) \quad (13)$$

Here, r,s are constants and ℓ_k and k^* are defined as in Theorem 1.

Proof can be found in the supplemental material[1].

Comparing Theorems 1 and 3: Both these theorems share important similarities. First, the regret is $O(\log T)$. Second, as the distance between clusters increase, the regret decreases. Third, the number of terms in the summation equals the number of arms in the optimal cluster + the number of sub-optimal clusters. However, there are significant differences. First, the distance between two clusters in Theorem 3 is measured differently: it is now the difference between the mean rewards of the best arms in the clusters. Second and more importantly, the constants involved in the bound of Theorem 3 arise from the results of (Kocsis and Szepesvári, 2006) and are significantly bigger than those involved in the bound of Theorem 1. We emphasize that the Theorem establishes only an upper bound on the regret: the actual regret for typical instances that arise in practise can be a lot smaller than this upper bound.

5. HLINUCBC for Contextual Bandit

A well known solution for the contextual bandit with linear payoffs is LINUCB (Li et al., 2010b) where the key idea is to apply online ridge regression to the training data to estimate the coefficients θ_k. We propose HLINUCBC (Algorithm 2) which extends this idea with both historical and clustering information.

Clustering Information: HLINUCBC deals with arms that are clustered; it applies online ridge regression at the per-cluster level to obtain an estimate of the coefficients $\hat{\theta}_i^c$ for each cluster (Line 7), plays the best cluster and then applies online ridge regression to the arms within the chosen cluster to obtain an estimate of the coefficients $\hat{\theta}_k$ for each arm (Line 6). To find the best cluster, at each trial t, HLINUCBC computes the quantities:

$$p_{t,i}^c \leftarrow \hat{\theta}_i^{c\top} x_t + \alpha \sqrt{x_t^\top (\mathbf{A}_i^c)^{-1} x_t} \quad (14)$$

$$p_{t,k} \leftarrow \hat{\theta}_k^\top x_t + \alpha \sqrt{x_t^\top (\mathbf{A}_k)^{-1} x_t} \quad (15)$$

For each cluster i and selects the cluster with the highest value of $p_{t,i}^c$ (Line 4). This quantity encapsulates the estimated reward from this cluster, along with an uncertainty term. Then, HLINUCBC finds the best arm within this cluster by computing for each arm k in this cluster, the quantity (15) and selects the arm with the highest value $p_{t,k}$ (Line 5).

Historical Information: HLINUCBC applies online ridge regression to the historical data at both per-cluster and per-arm levels. The aim is to collect enough information about how the context vectors and rewards relate to each other for each cluster and arm using the historical data, so that it can jump-start the algorithm by achieving a low number of errors at the early stages of the online exploration. At the initialization step, HLINUCBC is seeded with $\mathbf{H}_i^c$ and $\mathbf{H}_k$ which are respectively the history matrices for the clusters and arms. This is in contrast to LINUCB, where the initialization is done using an identity matrix. It is also seeded with the vectors b_i^{hc} and b_k^h, which record the weighted sum of historical contexts, with the weight being the rewards. In contrast, in LINUCB, this quantity is initialized to $\mathbf{0}$.

Algorithm 2 HLINUCBC

Input: $\alpha \in \mathbb{R}_{>0}$, history matrices $\mathbf{H}_i^c$ for each $i \in \{1,...,C\}$, $\mathbf{H}_k$ for each $k \in \{1,...,K\}$, weighted sum of contexts b_i^{hc} and b_k^h.

1: **for** arm $k \in \{1,...,K\}$ **do** $\mathbf{A}_k \leftarrow \mathbf{H}_k$, $b_k \leftarrow b_k^h$
2: **for** cluster $i \in \{1,...,C\}$ **do** $\mathbf{A}_i^c \leftarrow \mathbf{H}_i^c$, $b_i^c \leftarrow b_i^{hc}$
3: **for** $t \in \{1,...,T\}$ **do**
4: Choose cluster $i(t) = \text{argmax}_i p_{t,i}^c$
5: Play arm $k(t) = \text{argmax}_{k|c(k)=i(t)} p_{t,k}$
6: $\mathbf{A}_{k(t)} \leftarrow \mathbf{A}_{k(t)} + x_t x_t^\top$, $b_{k(t)} \leftarrow b_{k(t)} + r_{k(t)}(t) x_t$, $\hat{\theta}_{k(t)} \leftarrow \mathbf{A}_{k(t)}^{-1} b_{k(t)}$
7: $\mathbf{A}_{i(t)}^c \leftarrow \mathbf{A}_{i(t)}^c + x_t x_t^\top$, $b_{i(t)}^c \leftarrow b_{i(t)}^c + r_{k(t)}(t) x_t$, $\hat{\theta}_{i(t)}^c \leftarrow \mathbf{A}_{i(t)}^{c-1} b_{i(t)}$

Theorem 4. *With probability $1-\delta$, where $0 < \delta < 1$, the upper bound on the R(T) for the HLINUCB in the contextual bandit problem, K arms and d features (context size) is as follows:*

$$R(T) \leq \sigma \left(\sqrt{d \log \left(\frac{det(\mathbf{A}_T)^{1/2}}{\delta \, det(\mathbf{H})^{1/2}} \right)} + \frac{||\theta^*||}{\sqrt{\phi}} \right) \sqrt{8 T \log \left(\frac{det(\mathbf{A}_T)}{det(\mathbf{H})} \right)}$$

with $||x_t||_2 \leq L$ and $\phi \in R$

Theorem 4 shows that the HLINUCB upper bound has $log(\frac{det(\mathbf{A}_t)}{det(\mathbf{H})})$ under the square root whereas LINUCB has $log(det(\mathbf{A}_t)$. Recall from (Abbasi-Yadkori, Pál, and Szepesvári, 2011) that LINUCB has a regret guarantee which is almost the same as the one in Theorem 4 except for the $det(\mathbf{H})$ term. We now demonstrate that $\frac{det(\mathbf{A}_t)}{det(\mathbf{H})} \leq det(\mathbf{A}_t)$ and thereby show that HLINUCB has a provably better guarantee than LINUCB. The matrix H can be written as $I + D_h$ where I is the identity matrix and D_h is the design matrix constructed using the contexts in history. Both I and D_h are real symmetric and hence Hermitian matrices. Further, D_h is positive semi-definite since $D_h = \sum_i x_i x_i^T$, where the x_i are the historical contexts; to see this, note that $\forall y$, $y^T D_h y = \sum_i y^T x_i x_i^T y = \sum_i (x_i^T y)^2 \geq 0$. Since all the eigenvalues of I equal 1 and since all the eigenvalues of D_h are non-negative, by Weyl's inequality in matrix theory for perturbation of Hermitian matrices (Thompson and Freede, 1971), the eigenvalues of H are lower bounded by 1. Hence $\mathbf{det}(\mathbf{H})$ which is the product of the eigenvalues of H is lower bounded by 1 which proves our claim.

[1] https://tinyurl.com/y36y9hl8

6. META Algorithm

Intuitively, HUCBC can be expected to outperform HUCB *when* the quality of clustering is good. However, when the clustering quality is poor, this is far less likely to be the case. In order to choose between these two strategies correctly, we employ the META algorithm. Specifically, for the classical bandit problem, META uses UCB to decide between HUCBC and HUCB *at each iteration*. For the contextual bandit problem, META uses UCB to decide between HLINUCBC and HLINUCB *at each iteration*. The basic idea is that if the clustering quality is poor, the fraction of the time when META plays the strategy with both clustering and history will become vanishingly small, leading to the desired behavior.

Theorem 5. *The META algorithm is asymptotically optimal for the classical bandit problem under the assumption that the drift conditions of (Kocsis and Szepesvári, 2006) hold for HUCBC.*

Theorem 5 uses the assumption that HUCBC also satisfies the drift conditions like HUCB. In Theorem 1, we show that HUCB satisfies the drift conditions; this provides some evidence to support the assumption that HUCBC also satisfies them – however, we defer the formal proof of this hypothesis to future work.

7. Experimental Evaluation

Experiments with Synthetic Classical Bandit: We compare our proposed strategies HUCBC (Section 4) and META (Section 6) to the state-of-the-art competitors UCB (Auer, Cesa-Bianchi, and Fischer, 2002), HUCB (Shivaswamy and Joachims, 2012), and UCBC (Pandey, Chakrabarti, and Agarwal, 2007). HUCB is the degenerate version of HUCBC where all arms belong to the same cluster, i.e., no use of clustering information is made only historical data is employed. In contrast, UCBC is the degenerate version which uses the clustering information but no historical data.

Our synthetically generated data assumes 10 clusters and 100 arms. A random permutation assigns each arm k to a cluster $c(k)$, where the centroid of cluster i is defined as $\lambda(i) = \frac{1}{2}\left(u_i + \frac{1}{i}\right)$ and $u_i \sim \mathcal{U}(0,1)$. The reward obtained when arm k is played is sampled from $\mathcal{U}(0, 2\alpha_k \lambda(c(k)))$, where α_k is a arm dependent constant. Thus, the expected reward for cluster i is $\lambda(i)$. The historical data is generated as follows: for 25% we generate data playing them X times where X is sampled from a Poisson distribution with parameter 10. We report the results of 20 trial in in Figure 1a. Each trial consisted of 10^4 rounds and we plot the the number of rounds per-round-reward (cumulative reward until that round / number of rounds). The error bars correspond to the interval of ± 1 standard deviation around the mean. Clearly, the per-round-reward of HUCBC is the fastest to converge to that of the best arm. Interestingly, while historical data improves the performance of HUCB over UCB only to a small degree (possibly due to asymmetry in history), combining this information with the clustering information improves the performance of HUCBC over UCBC to a significantly larger degree. This is an effect of the fact that even though historical data is sparse and asymmetric at the arm level, combining this information at the cluster level is more impactful. META converges rather quickly to the correct choice (HUCBC) in this setting.

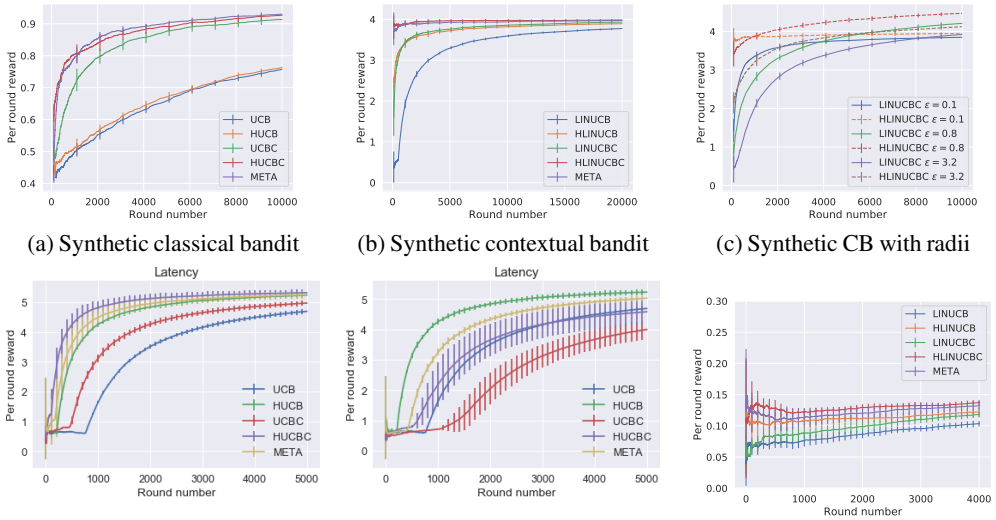
Experiments with Synthetic Contextual Bandit: We will now compare our proposed contextual bandits HLINUCBC (Section 5) and the META algorithm selection to LINUCB (Li et al., 2010b), HLINUCB and LINUCBC. Similarly to HUCB and UCBC, HLINUCB and LINUCBC are degenerated versions of HLINUCBC using either only the historical data or the clustering information.

The synthetic data was created by assuming 10 clusters and 100 arms. Again, a random permutation assigns each arm k to a cluster $c(k)$. The centroid θ_i^c of cluster i is sampled from $\mathcal{N}(\mathbf{0}, I_5)$. The coefficient θ_k of arm k is defined as $\theta_k = \theta_i^c + \epsilon\nu_k$, where ν_k is fix but sampled from $\mathcal{N}(\mathbf{0}, I_5)$. The reward for playing arm k in context x is sampled from $\mathcal{U}(0, 2\theta_k^\top x)$. Thus, our synthetically generated problem is linear in the input and the expected distance between clusters is $\sqrt{5}\epsilon$. By varying ϵ, we control the tightness of the clusters and can observe the impact of clustering information with varying quality of clusters. Contexts in each round of a trial are drawn i.i.d from the multivariate normal distribution $\mathcal{N}(\mathbf{0}, I_5)$. We created asymmetric historical data for arms by playing an arm X times with X distinct contexts, where $X \sim Poisson(10)$.

We report the results for $\epsilon = 0.1$ in Figure 1b. As in the case of classical bandits, we repeat the experiment 20 times, where each trial consists of 10,000 rounds. The error bars correspond to the interval of ± 1 standard deviation around the mean. HLINUCBC is clearly outperforming its competitors and META is able to identify HLINUCBC as the stronger method over HLINUCB. In this set up, historical data and clustering are equally important such that HLINUCB and LINUCBC provide similar results. LINUCB clearly provides the worst results. We also compare the performance of HLINUCBC vs. LINUCBC under different values of $\epsilon \in \{0.1, 0.8, 3.2\}$. We normalize the rewards in these three distinct settings to enable a meaningful comparison. We present the results in Figure 1c. Since HLINUCBC utilizes historical data in addition to the clustering information, we can see that it improves upon the performance of LINUCBC for every setting of ϵ.

Experiments with Real-World Data: We compare our proposed bandit algorithms on two different real-world problems. The classical bandits are compared on the task of latency-based web server selection while the contextual bandits are analyzed for the Warfarin drug dosage problem.

Latency-based Web Server Selection: (Krishnamurthy, Wills, and Zhang, 2001b) Essentially one needs to decide from what source one should pull content available at multiple sources while minimizing the cumulative latency for successive retrievals. Similar to (Vermorel and Mohri, 2005), we assume that only a single resource can be picked



(a) Synthetic classical bandit

(b) Synthetic contextual bandit

(c) Synthetic CB with radii

(d) Web server selection with good clustering

(e) Web server selection with poor clustering

(f) Warfarin dosage

Figure 1. (a) HUCBC outperforms its competitors. META quickly learns to follow HUCBC (b) HLINUCBC outperforms its competitors. META quickly learns to follow HLINUCBC (c) Rewards for the different cluster-based methods under different cluster radii ϵ. HLINUCBC benefits from historical data in all cases. (d) HUCBC provides significantly better rewards than competitors when clustering is good. Meta learns to follow HUCBC (e) HUCB outperforms HUCBC when clustering is poor and META learns to follow HUCB (f) HLINUCBC outperforms competitors and META learns to follow this

at a time. The university web page data set[2] features more than 700 sources with about 1300 response times each. In order to facilitate our clustering approach in this setting, we perform two types of clustering: 1) split the resources based on average latencies into five categories ranging from 'fast' to 'slow', 2) based on domain names into 17 clusters. Historical data was generated by choosing 200 observations in total at random. In Figures 1d and 1e we show the mean cumulative reward for the UCB, UCB with history (HUCB), UCB with clustering (UCBC), UCB with both (HUCBC), and META over 10 repetitions. Before each experiment, the data has been shuffled. We can observe a clear impact of employing clustering and history and the combination of both, and the type of clustering. The first type of clustering achieves an effective grouping of arms while the second one does not. Confirming our observations on the synthetic data, only clustering information provides better results than only historic data when clustering is of good quality. The two kinds of clustering also highlight the usefulness of our META approach that in both cases converges to the correct approach.

Warfarin Drug Dosage: The Warfarin problem data set is concerned with determining the correct initial dosage of the drug Warfarin for a given patient. Correct dosage is variable due to patient's clinical, demographic and genetic context. We select this benchmark because it enables us to compare the different algorithms since this problem allows us to create a hierarchical classification data set. Originally, the data set was converted to a classification data set with three classes using the nominal feedback of the patient. We create a hierarchy as follows. We divide each of the original three classes into five more granular classes. Bandit methods which do not use hierarchy will simply tackle this problem as a classification task with 15 classes. Others will first assign instances to one of the three classes and finally decide to choose one of the final five options. The order of the patients is shuffled, 1500 of them are chosen as historical data. We report the mean and standard deviation of 10 runs in Figure 1f. HLINUCBC is outperforming all competitor methods and also META is able to successfully detect that HLINUCBC is the dominating method. For this problem HLINUCB provides better results than LINUCBC, indicating that historic data is worth more than clustering information. All methods clearly outperform LINUCB.

8. Conclusions

We introduced a variety of algorithms for classical and contextual bandits which incorporate historical observations and pre-clustering information between arms in a principled manner. We demonstrated their effectiveness and robustness both through rigorous regret analysis as well as extensive experiments on synthetic and real world datasets. Two interesting open problems emerge from this work: 1) Are there instance independent regret bounds for HUCBC which depend only on the total number of clusters and the number of arms in the optimal cluster? 2) What are the upper and lower bounds on the regret of HLINUCBC?

[2]https://sourceforge.net/projects/bandit/

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.

Agrawal, S., and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *ICML (3)*, 127–135.

Alon, N., and Spencer, J. H. 2016. *The Probabilistic Method*. Wiley Publishing, 4th edition.

Auer, P., and Cesa-Bianchi, N. 1998. On-line learning with malicious noise and the closure algorithm. *Ann. Math. Artif. Intell.* 23(1-2):83–99.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.* 32(1):48–77.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3):235–256.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. E. 2011. Contextual bandits with linear payoff functions. In Gordon, G. J.; Dunson, D. B.; and Dudik, M., eds., *AISTATS*, volume 15 of *JMLR Proceedings*, 208–214. JMLR.org.

Gentile, C.; Li, S.; Kar, P.; Karatzoglou, A.; Zappella, G.; and Etrue, E. 2017. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 1253–1262.

Gentile, C.; Li, S.; and Zappella, G. 2014. Online clustering of bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 757–765.

Kaufmann, E.; Korda, N.; and Munos, R. 2012. Thompson Sampling: An Asymptotically Optimal Finite Time Analysis. In *Algorithmic Learning Theory, Proc. of the 23rd International Conference (ALT)*, volume LNCS 7568, 199–213. Lyon, France: Springer.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *ECML*, volume 6, 282–293. Springer.

Korda, N.; Szörényi, B.; and Li, S. 2016. Distributed clustering of linear bandits in peer to peer networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 1301–1309.

Krishnamurthy, B.; Wills, C.; and Zhang, Y. 2001a. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 169–182. ACM.

Krishnamurthy, B.; Wills, C.; and Zhang, Y. 2001b. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, 169–182. New York, NY, USA: ACM.

Lai, T. L., and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6(1):4–22.

Langford, J., and Zhang, T. 2007. The epoch-greedy algorithm for multi-armed bandits with side information. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *NIPS*. Curran Associates, Inc.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010a. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 661–670. New York, NY, USA: ACM.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010b. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, 661–670. USA: ACM.

Liu, Y.; Li, Z.; Xiong, H.; Gao, X.; and Wu, J. 2010. Understanding of internal clustering validation measures. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, 911–916. Washington, DC, USA: IEEE Computer Society.

Maillard, O., and Mannor, S. 2014. Latent bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 136–144.

Nguyen, T. T., and Lauw, H. W. 2014. Dynamic clustering of contextual multi-armed bandits. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, 1959–1962.

Pandey, S.; Chakrabarti, D.; and Agarwal, D. 2007. Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th international conference on Machine learning*, 721–728. ACM.

Sharabiani, A.; Bress, A.; Douzali, E.; and Darabi, H. 2015. Revisiting warfarin dosing using machine learning techniques. *Computational and mathematical methods in medicine* 2015.

Shivaswamy, P., and Joachims, T. 2012. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics*, 1046–1054.

Tang, L.; Rosales, R.; Singh, A.; and Agarwal, D. 2013. Automatic ad format selection via contextual bandits. In *CIKM*.

Thompson, R. C., and Freede, L. J. 1971. On the eigenvalues of sums of hermitian matrices. *Linear Algebra and Its Applications* 4(4):369–376.

Thompson, W. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25:285–294.

Vermorel, J., and Mohri, M. 2005. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning*, 437–448.

Wang, Z.; Zhou, R.; and Shen, C. 2018. Regional multi-armed bandits. *arXiv preprint arXiv:1802.07917*.